

TECHNIQUES FOR NETWORK TRAFFIC ENGINEERING

Field of the Invention

The present invention relates generally to communication over networks, and, more particularly, to determining effective paths through networks.

5

Background of the Invention

Traffic engineering (TE) systems attempt to control a network to provide an optimal usage of the network for the particular traffic demands experienced. A TE system in a connection-oriented network, where connections are made between source and destination nodes, can be generally classified as online or offline.

With an online TE system, connection requests are generally assumed to arrive one at a time. For each connection request, an online TE system typically computes each path independently without assuming knowledge of future requests. One common technique used to compute a path is shortest path first (SPF), which computes the shortest path between a source node and destination node. Another common technique is constrained SPF (CSPF), which first prunes a network so that links not meeting a bandwidth requirement for a connection request are removed, then the shortest path is chosen using the pruned network. After a path is determined, the connection is usually set up by signaling. The total provisioning time from when a connection request arrives to when a connection is established may take tens to hundreds of milliseconds. Thus, path determination in an online TE system must be relatively fast, e.g., on the order of milliseconds. Techniques such as CSPF can be “greedy” in that the path determined may be quite long and therefore use a large amount of resources in the network.

With an offline TE system, typically the aggregate behavior of all connection requests is assumed to be known before the corresponding paths are computed. Using topology information, link capacities and a traffic demand matrix, a centralized server typically performs global optimization to determine the path for each connection request. Once path design is completed, the connections are typically set up by a network management system. It is generally agreed that an offline system with global optimization can achieve considerable improvement in resource

utilization over an online system, if the traffic matrix accurately reflects the real load the network will be required to carry. It is not generally understood how much benefit can be derived from an offline system that uses inaccurate load information. A need therefore exists to quantify the benefits of offline TE for operational networks in the
5 absence of exact load information. In addition, the total pre-provisioning time with an offline system typically takes minutes or hours. Another problem with offline TE systems is that they typically need accurate demand information for the resultant paths to be accurate.

A need therefore exists to improve offline path computation
10 techniques, with or without accurate demand information, for optimization in an online TE system.

Summary of the Invention

The present invention provides techniques for improved traffic
15 engineering (TE) in connection-oriented networks, such as multiprotocol label switching (MPLS) or generalized MPLS (GMPLS). Techniques for online, offline, and combinations of online and offline computation techniques are disclosed.

In a first aspect of the invention, in response to a request, it is determined whether any path of a number of predetermined paths meets requirements
20 corresponding to the request. The predetermined paths are determined by substantially maximizing carried demand in a network using at least traffic demand estimates and network topology information and by performing routing for the substantially maximized carried demand. The carried demand is generally a total amount of demand that can be carried in the network. If a given path meeting the
25 requirements is found, a connection utilizing the given path is created, if possible. The predetermined paths are computed offline and routing that uses only the predetermined paths is referred to herein as offline design-based routing (DBR). The requirements for the path may be bandwidth and other constraints.

The traffic estimates may be determined using previously measured
30 traffic demands or historical traffic demands or both, and the network topology may be determined by using information from link-state routing.

There are a number of techniques suitable for performing the substantial maximization and the routing. For instance, substantial maximization may be performed by obtaining a threshold that maximizes a number of connections that can be accepted and adjusting traffic demand for each of a number of node pairs in the network until the carried demand is substantially maximized. As another example, routing may be performed by minimizing a total bandwidth-length product subject to a number of constraints, such as edge capacity constraints and path-assignment constraints.

Additionally, offline DBR techniques can utilize uncertain static demand information. The path-assignment constraints may therefore comprise constraints that constrain a sum of an amount of demand in units of path capacity to be greater than a first product. The first product is a multiplication between a threshold and a second sum. The second sum is a sum of an average demand and a second product. The second product is a multiplication between a number indicating a distance from a standard deviation and a standard deviation of a normal distribution function.

An offline DBR technique may also utilize dynamic connections. Routing therefore may comprise minimizing a total bandwidth-length product subject to a number of constraints including edge capacity constraints. The edge capacity constraints may include constraining end-to-end blocking probability for a node pair to be less than a predetermined amount. As another example, routing may comprise determining a threshold minimum capacity assigned for a node pair that will meet a given blocking probability.

In yet another aspect of the invention, an offline DBR technique is combined with one or more adaptive DBR techniques, which can compute a path online should a path computed offline be deemed unsuitable for use. The adaptive DBR techniques may include determining whether a designed load between source and destination nodes is greater than a measured load between the source and destination nodes. If so, edges that do not have a first available bandwidth are pruned from the network topology. If not, edges that do not have a second available bandwidth are pruned from the network topology. The available bandwidths include, but are not limited to, trunk reservations. A first pruned network is created by the

pruning. If a path meeting the requirements is found, a connection on the path is set up, if possible. If no path is found, then edges that do not have a first available bandwidth are pruned from the first pruned network to create a second pruned network, a shortest path from the source node to the destination node in the second pruned network is determined, and a connection on the shortest path is attempted to be created. Generally, the first available bandwidth is zero and the second available bandwidth is a predetermined trunk reservation.

In another aspect of the invention, a modification of a conventional constrained shortest path first (CSPF) technique is presented. In this exemplary online computation technique, a first shortest path between a source node and destination node is determined. Edges not having a first available bandwidth are pruned from the network, thereby creating a first pruned network. A second shortest path is computed between the source node and the destination node using the first pruned network. If a length of the second shortest path is equivalent to a length of the first shortest path, a connection on the second shortest path is created, if possible. If the length of the second shortest path is not equivalent to a length of the first shortest path, then edges not having a second available bandwidth are pruned from the first pruned network, thereby creating a second pruned network. A third shortest path is computed between the source node and destination node using the second pruned network. A connection on the third shortest path is created, if available.

Brief Description of the Drawings

FIG. 1 is a block diagram of a communication system in accordance with a preferred embodiment of the invention;

FIG. 2 illustrates different classifications for routing;

FIG. 3 is a block diagram illustrating a framework for offline design based routing (DBR), which can be used in adaptive DBR systems;

FIG. 4 illustrates a method for offline path optimization for network traffic engineering;

FIG. 5 illustrates an example network topology used to show a benefit of path optimization for network traffic engineering;

FIG. 6 shows pseudocode of an exemplary method for adaptive DBR, which can combine online and offline network traffic engineering; and

FIG. 7 shows pseudocode of an exemplary method for online traffic engineering using a modified constrained shortest path first (CSPF) with trunk requirements (TRs) technique.

Detailed Description

For ease of reference, the present disclosure is divided into the following sections: Introduction; Path Design with DBR; and Routing With Resource Management.

Introduction

Referring now to FIG. 1, a communication system 100 is shown. Communication system 100 comprises a DBR server 110 coupled to a network 130. Network 130 comprises a source node 120, intermediate nodes 150-1 through 150-5, edges 180-1 through 180-10, and a destination node 140. DBR server 110 comprises a processor 111, memory 112 and network interface 113. The source node 120, destination node 140, and intermediate nodes 150 could be routers, switches, or other types of network elements. For purposes of exposition, the source node 120, destination node 140, and intermediate nodes 150 will be considered to be label switch routers (LSRs) or have an LSR portion (not shown) in addition to other portions. Memory 112 comprises a DBR database 114. DBR database 114 comprises a number of paths 115, each path 115 having a predetermined DBR path, source, destination, and bandwidth. The processor 111 executes one or more programs (not shown) in order to implement the techniques of the present invention. It should be noted that FIG. 1 only shows one source and one destination node for simplicity. In general, there will be a multiplicity of source and destination nodes.

In general, the offline design stage 101 is used to determine, using techniques described in detail below, paths 115 from a source node 120 to a destination node 140. This determination is performed offline. Paths 115 usually include bandwidth requirements. The paths 115 that are determined offline are used during the online provisioning stage 102. Broadly, a source node 120 desires, based

on a connection request 105 from a client (not shown), a connection to another client (not shown) via the destination node 140. The connection request 105 usually comprises a source, destination, and bandwidth requirement, and may comprise additional constraints. The LSR portion of source node 120 then sends a path request 122 to the DBR server 110, requesting the “best” path to the destination node 140. The path request 122 generally includes the source, destination, and bandwidth requirement from the connection request 105. The DBR server 110 acts to comply with the path request 122 by determining a path in paths 115 that meets the requirement of the path request 122. The DBR server 110 sends a reply 123 to the source node 120. The reply generally includes a DBR path, such as DBR path 160. The path request 122 and reply 123 usually travel through network 130 or through a separate signaling network (not shown).

In an aspect of the invention, the LSR portion of the source node 120 will only use paths 115 that are predetermined through offline computation techniques. Upon receiving the reply 123, the LSR portion of the source node 120 will then attempt to setup the DBR path (e.g., DBR path 160) from the source node 120 to the destination node 140. The connection setup may be made through any suitable technique, such as through the RSVP-TE signaling protocol as described in RFC 3209, “RSVP-TE: Extensions to RSVP for LSP Tunnels,” (December 2001), the disclosure of which is hereby incorporated by reference. In the example of FIG. 1, the DBR path 160 is set up over edges 180-1, 180-3, 180-4 and 180-5. If there is no suitable path in paths 115, the DBR server 110 can send a reply 123 that includes a rejection of the path request 122. The offline computation techniques are referred to herein as offline DBR techniques.

In another aspect of the invention, the LSR portion of the source node 120 first tries to use any paths 115 determined through offline DBR techniques but if the path is deemed unsuitable, can fall back to an online computation technique in order to determine an alternate path. For example, the constrained shortest path first (CSPF) path 170 may be determined and set up over edges 180-10, 180-6, and 180-5 if the DBR path 160 is deemed unsuitable. This is referred to as adaptive DBR herein and is explained in more detail below, for instance in reference to FIG. 6.

In another aspect of the invention, the LSR portion of the source node 120 uses only online TE techniques to respond to connection requests 105 and to create a path determined online. An example of an online TE technique is shown in reference to FIG. 7.

5 It should be noted that source node 120, destination node 140, and intermediated nodes 150 will also generally comprise a switch matrix and multiple network interfaces such as found in routers and switches. It should also be noted that edges 180 may be bidirectional. Additionally, network 130 may be any packetized or optical network design. Illustratively, network 130 may be network supporting
10 multiprotocol label switching (MPLS) or generalized MPLS (GMPLS).

 The techniques described herein may be implemented through hardware, software, firmware, or a combination of these. Additionally, the techniques may be implemented as an article of manufacture comprising a machine-readable medium, as part of memory 112 for example, containing one or more programs that
15 when executed implement embodiments of the present invention. For instance, the machine-readable medium may contain a program configured to perform some or all of the steps of the present invention. The machine-readable medium may be, for instance, a recordable medium such as a hard drive, an optical or magnetic disk, an electronic memory, or other storage device.

20 Turning now to FIG. 2, classifications of routing methods for TE are shown. The classifications are shown as a taxonomy, which is not intended to be exhaustive. However, FIG. 2 covers a broad array of schemes that make use of offline and online measurements, local and global optimizations, resource management schemes and signaling requirements. At a high level, the methods may
25 be classified according to whether offline design optimization is used. The design-based routing family is distinguished by the use of an estimated traffic demand matrix and global path optimization. The shortest-path routing family involves certain shortest path computations, with or without resource management, for one source-destination pair at a time. In the present disclosure, the term “source” includes a
30 source node, and the term “destination” includes a destination node. The routing algorithms in the shortest path family include the standard SPF and CSPF methods. In the present disclosure, an extension to this family is presented with resource

management, which is an improved CSPF technique with trunk reservation (TR). The improved technique is shown as CSPF_TR.

The routing algorithms in the design-based routing family may be further grouped based on their use of online measurements. In DBR, optimally
5 predetermined paths based on approximate knowledge of the demand matrix are used in setting all source-destination paths online. In the hybrid scheme, called adaptive DBR, online measurements are used to further improve path selection using admission control.

Turning now to FIG. 3, a framework 300 for offline DBR is shown.
10 Framework 300 may also be considered to be a method for offline DBR. The framework 300 comprises information from link-state routing 310, traffic demand estimate information 320, a path optimization module 330 with DBR, a DBR database 340, and an admission control module 350. Note that 310, 320, 330 and 340 reside at the DBR server 110, while 350 resides in each source node 120 (e.g., having an LSR).
15 The path optimization module 330 collects information from link-state routing 310 and traffic demand estimate information 320. The information from link-state routing 310 may comprise network topology, link capacities, and other relevant information such as constraints. The traffic demand estimate 320 is an estimate of traffic on a network.

20 Based on this input information, the optimization module 330 computes all optimized paths offline and stores the results in the DBR database 340 for future use. As described in more detail below, the DBR database 340 comprises a set of paths from each source node to each destination node and bandwidth allowed on each path. When an external connection request arrives from a source computer,
25 the admission control module 350 would try to set up a connection to a specified destination. In non-adaptive DBR, admission control is only applied on a DBR path found in the DBR database 340. In other words, if there is no previously determined path in the DBR database 340 meeting the requirements in the connection request, the admission control module 350 rejects the connection request. In adaptive DBR,
30 admission control is applied on a DBR as well as alternate (e.g., CSPF) paths. Consequently, in adaptive DBR, even though a path meeting the requirements in the

connection request cannot be found, the admission control may perform an online technique in order to meet the connection request.

Non-adaptive DBR is described immediately below and adaptive DBR is described in the Routing With Resource Management section below.

5

Path Design with DBR

Techniques for TE using offline DBR are now described.

A. Definitions and Notations

10 A network topology is represented by a graph $G(V, E)$, where V denotes the set of nodes (also called vertices) and E denotes the set of edges (also called links) and each edge, $e \in E$, has a capacity of C_e . It should be noted that a graph may be represented by any type of data, such as a linked list.

Length for a path may be a function of actual distance or other salient
15 parameters used in the optimization criteria. Focus herein is placed on the optical transport network, where the edges and the connections are typically bi-directional. However, the techniques herein are applicable to non-optical transport networks. Demands, in terms of the number of bandwidth, vary from one source-destination pair to another. For an optical transport network, for instance, each connection may be
20 considered to be a lightpath of the same size (e.g., one lambda).

Let K denote the set of all node pairs in the network. For static connections, the traffic demand, d_k , is expressed in terms of the total bandwidth of connections that need to be routed for node pair $k \in K$. For dynamic connections, the traffic demand, δ_k , is the offered load in Erlangs for node pair $k \in K$. Let P_k denote
25 the set of all paths for node pair k and let P denote the set of all paths for all node pairs; that is, $P = \bigcup_k P_k$. The output of path design determines the amount of demand that needs to be assigned to each path. Let the quantity x_p represent the amount of demand assigned to path $p \in P$. Finally, let Q_e denote the set of all paths that traverse edge e . The subsequent sections present DBR formulations for a case with
30 no protection (i.e., no restoration in case of edge failure).

B. DBR with Accurate Static Demand Information

A computer system implementing DBR determines all end-to-end paths concurrently and optimally using the network topology, edge capacities, traffic demand matrix and possibly other constraints. A path design objective is to minimize the total bandwidth-length product. Furthermore, the optimization problem is subject to meeting the designated end-to-end demands, from sources to destinations, while not exceeding the edge capacities. If all connections cannot be established in the existing network due the constraints implied by the fixed edges capacities, it is recommended that the problem be to be solved in two stages, as shown broadly in method 400 of FIG. 4. Method 400 is performed by a computer system, such as the traffic engineering server 110 shown in FIG. 1, in order to determine paths used later for routing connections. In Stage 1, step 410, a goal is to substantially maximize the carried demand in a network. The carried demand is the total amount of demand that can be carried in the network. It should be noted that substantially maximizing does not mean absolute maximization, although absolute maximization is beneficial. Thus, any technique that increases carried demand in a network is suitable for substantial maximization, although in general the closer to maximal the carried demand is, the better the overall result will be. In Stage 2, step 420, optimal routing is performed for the carried demand derived in Stage 1. These two stages may be formulated in different ways, as described in detail below.

One general formulation for Stage 1 is as follows:

P1(θ)

$$\max_{x_p, \theta_p} \sum_k \theta_k d_k \quad (1)$$

25

subject to

$$\sum_{p \in P_k} x_p \geq \theta_k d_k, \quad \forall k \in K \quad (2)$$

$$\sum_{p \in Q_e} x_p \leq C_e, \quad \forall e \in E \quad (3)$$

$$x_p \geq 0, \quad \forall p \in P \quad (4)$$

$$0 \leq \theta_k \leq 1, \quad \forall_k \in K \quad (5)$$

Having obtained the threshold θ_k^* that maximizes the number of connections that can be accepted, the traffic demand is adjusted for each node pair k as $d_k^* \leftarrow \theta_k^* d_k$. It should be noted that Stage 1 optimization can be performed with a common demand multiplier for all node-pairs, i.e., $\theta_k = \theta, \forall k$, which will ensure a fair admission policy amongst all node-pair demands at the expense of reducing the overall carried traffic. Note that the set of all paths for all node pairs P have been computed and qualified at the outset. For example, for each node pair, the set of possible paths may consist of all paths with length less than the length of the shortest path plus some threshold value. In general, using very long paths for multicommodity flow problems has diminishing utility.

A Stage 2 optimization problem is as follows:

15 **Q1 (λ * length)**

$$\min_{x_p, \lambda_e} \sum_{e \in E} \lambda_e l_e \quad (6)$$

Subject to

$$20 \quad \sum_{p \in P_k} x_p \geq d_k, \quad \forall k \in K \quad (7)$$

$$\sum_{p \in Q_e} x_p \leq \lambda_e, \quad \forall e \in E \quad (8)$$

$$\lambda_e \leq C_e, \quad \forall e \in E \quad (9)$$

$$x_p \geq 0, \quad \forall p \in P \quad (10)$$

25 Equation set (8) is an equation set having path-assignment constraints, while equation set (9) is an equation set having edge capacity constraints.

It can be shown that the total demand met in Q1 is the same as the objective value of problem P1, however, the output of Q1 provides the assignments of

demands to paths for each node pair (i.e., x_p) with least cost packing. It can also be shown that when the edge capacities are large, the effect of performing this optimization is simply to let each node pair use the shortest path.

In this case, no concurrent optimization of all demands is necessary.

- 5 Indeed, Q1 which is minimization of the total bandwidth-length product subject to edge capacity constraints, can be thought of as a simple non-trivial extension of the shortest path routing when there are capacity constraints on the edges.

The output of this integer linear program (ILP), or its rounded linear program (LP) relaxation, gives the paths and their allocated flows for each node pair. 10 Clearly, the total bandwidth-length product of any other scheme is at best equal to the optimum value obtained above for a prescribed demand matrix. In particular, this is true for any online TE system, where each connection request is processed one at a time at each arrival, such as in SPF and CSPF. It can be shown that the optimum value determined through the above scheme is better than an online TE system.

- 15 In SPF, static edge weights are used for path computation, where the weight could be a function of distance, inverse of edge capacity ($1/C_e$), or other parameters. Arriving requests are continuously assigned to the computed paths although some edges may become saturated, resulting in blocking, while other edges may still be underutilized. To alleviate this lack of load balancing inherent in SPF, 20 CSPF can use dynamic weights, such as the residual capacity (e.g., $1/(C_e - \lambda_e)$) that reflects the current availability of the edge resources. CSPF clearly dominates SPF in many circumstances, but its utility vanishes rapidly as the total demand in the network increases. To summarize, if $v(policy)$ denotes the optimal objective (such as carried demand) of a routing policy, then $v(DBR) \geq v(CSPF)$.

25

C. Value of Traffic Demand Information

- It may be argued that the lack of optimality of online path computation is purely due to the non-optimal arrival sequence of connection requests in time. That is, if one could examine all possible permutations of arrival sequences, the optimal 30 solution will simply follow by applying SPF/CSPF to that specific sequence. To see why solutions based on the optimal arrival sequence cannot match the solution to Q1,

consider the simple example shown in FIG. 5. Here, each edge has capacity one unit (λ) and length, or cost, as indicated in the figure. There are two connection requests, each needing one unit of capacity, between A and F to be routed. Observe that for SPF and CSPF, the paths are (A, C, E, F) and one of (A, B, F) or (A, D, F).
 5 Thus the total cost for SPF or CSPF is 14. However, the optimal paths would take (A, B, E, F) and (A, C, D, F), resulting in the total cost of 6. Furthermore, notice that no permutation of the arrival sequences in SPF or CSPF will result in the optimal paths. It can be concluded that the concurrency of path assignments afforded by optimization is a stronger factor for the best use of network resources than the order of arrivals or
 10 load balancing (e.g., CSPF). In other words, use of the traffic demand information and concurrent assignment of demands to routes enables better packing of the demands in the network in ways that even the best sequential solution, with or without load balancing, cannot match.

This concurrency in the design of paths (as enabled by DBR of the
 15 present invention) turns out to be an even more important factor for the effective use of network resources than the network load itself.

This is true whether load is interpreted in the static case with or without perfect information or in the dynamic case. The next section defines DBR in the context of uncertain traffic demand and dynamic connections.

20

D. DBR with Uncertain Static Demand Information

It is commonly argued that exact end-to-end demand information is almost never available in practice. Often, this argument is used erroneously to conclude that the class of purely online path computation schemes is the only suitable
 25 solution in practice, since this class does not require information about the end-to-end traffic demand matrix. It can be shown that, using techniques of the present invention, even demand information with considerable uncertainty can be utilized to give a better path design than can be computed by SPF or CSPF. To explore the uncertainty issue further, assume that $\tilde{d}_k \in N(\bar{d}_k, \sigma_k)$ and reformulate *PI/QI* when
 30 end-to-end demands are no longer fixed but are random variables. The direct translation of the stochastic variant of *PI* using point estimates of the random variable

$d_k, \forall k$ would not lead to the correct optimization model, however. This is because the carried load in a stochastic setting is the minimum of (i) the provisioned bandwidth and (ii) the offered load; in other words, $\min\{\sum_{p \in P_k} x_p, \sum_k \theta_k \tilde{d}_k\}$. The condition set from equation set (2) with point estimates for d_k only ensures that the mean provisioned demand exceeds the offered load. Therefore, this condition can be substituted with a more stringent requirement that the offered load is met with a high probability. Thus, Stage 1 optimization (i.e., step 410 of FIG. 4) in this setting becomes the following:

10

 $\tilde{\mathbf{P}} \mathbf{1}(\theta)$

$$\max_{x_p, \theta_k} E\{\sum_k \theta_k d_k\} \quad (11)$$

subject to

15

$$\Pr\{\sum_{p \in P_k} x_p \geq \theta_k d_k\} \geq 1 - \epsilon_k, \quad \forall k \in K \quad (12)$$

$$\sum_{p \in Q_e} x_p \leq C_e, \quad \forall e \in E \quad (13)$$

$$x_p \geq 0, \quad \forall p \in P \quad (14)$$

$$0 \leq \theta \leq 1, \quad \forall k \in K \quad (15)$$

20 Equation set (13) is an equation set having edge capacity constraints.

It is noted that the constraint set in equation set (2) has been replaced with the equation set in equation set (12) which ensures, with arbitrarily high probability $1 - \epsilon_k$, that the optimized fraction of the offered load is met. It is also noted that because of this constraint, the carried load, namely $\min\{\sum_{p \in P_k} x_p, \sum_k \theta_k \tilde{d}_k\}$, is equal to $\sum_k \theta_k \bar{d}_k$ which is the objective maximized.

25

Given that $d_k \sim N(\bar{d}_k, \sigma_k) \forall k$, the constraint set (12) can be simplified as

$$\sum_{p \in P_k} x_p \geq \theta_k (\bar{d}_k + \eta_k \sigma_k) \quad (16)$$

where

$$\eta_k = \sqrt{2} \text{erf}^{-1}(1 - 2\epsilon_k) \quad (17)$$

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2} dy \quad (18)$$

Equation set (16) is an equation set having path-assignment constraints.

It can be observed that over a long series of the realizations of the random demands, the above procedure will outperform any online scheme with the same requirements. In particular: $E\{v(DBR)\} \geq E\{v(CSPF)\}$.

As in the deterministic formulation, better packing of the flows x_p to paths p can be obtained from the equivalent model to $Q1$ obtaining θ^* from $P2$:

Q2 (λ * length)

$$\min_{x_p, \lambda_p} \sum_{e \in E} \lambda_e l_e \quad (19)$$

subject to

$$\sum_{p \in P_k} x_p \geq \theta_k^* (\bar{d}_k + \eta_k \sigma_k) \quad \forall k \in K \quad (20)$$

$$\sum_{p \in Q_e} x_p \leq \lambda_e, \quad \forall e \in E \quad (21)$$

$$\lambda_e \leq C_e, \quad \forall e \in E \quad (22)$$

$$x_p \geq 0, \quad \forall p \in P \quad (23)$$

It is noted that $Q2$ is a deterministic optimization problem and is feasible due to the rescaling of demands derived from $P2$.

E. DBR with Dynamic Connections

When connections are dynamic, there is always a chance of blocking even with generous dimensioning of the paths. In this case, the DBR objective is to minimize network resource usage subject to meeting the traffic demand up to a given level of blocking probability. Variants of this problem have been considered in D. Mitra and I. Ziedins, "Virtual Partitioning by Dynamic Priorities: Fair and Efficient Resource-Sharing by Several Services," in Proc. Int'l Zurich Seminar on Digital Communications, Lecture Notes in Comp. Sci., Broadband Communication Link, B. Plattner (editor), Springer, 173-185 (1996), the disclosure of which is hereby incorporated by reference, where three approaches were proposed: complete partitioning, complete sharing and virtual partitioning. For simplicity and clarity of the presentation, only complete partitioning will be described. However, DBR techniques disclosed herein are extensible to all three approaches.

Assume that connection requests between node pair k arrive according to a Poisson process with offered load δ_k , and a logical path with capacity $L_k (= \sum_{p \in P_k} x_p)$ is assigned for node pair k . Then the end-to-end blocking probability for node pair k is given by the Erlang B formula

$$B(\delta_k, L_k) = \frac{\delta_k / L_k!}{\sum_{i=0}^{L_k} \delta_k / i!} \quad (24)$$

20

The optimization procedure for DBR with dynamic connections can be formulated as follows:

$$\begin{aligned} & \text{Q3}(\lambda * \text{length}) \\ & \min \sum_{e \in E} \lambda_e \end{aligned} \quad (25)$$

25

subject to

$$B(\delta_k, L_k) \leq \beta, \quad \forall k \in K \quad (26)$$

$$\sum_{p \in P_k} x_p \geq L_k, \quad \forall k \in K \quad (27)$$

$$\sum_{p \in Q_e} x_p \leq \lambda_e, \quad \forall e \in E \quad (28)$$

$$\lambda_e \leq C_e, \quad \forall e \in E \quad (29)$$

$$x_p \geq 0, \quad \forall p \in P \quad (30)$$

5

Note that the non-linear constraint of equation set (26) can be eliminated by first solving for L_k . In this case, the optimization problem can be simplified by using the following procedure:

10

Q3 (λ * length)

1. Compute $L_k^* = \arg \min_{L_k} \{B(\delta_k, L_k) \leq \beta\}, \forall k \in K$.
2. Solve Q1 with $d_k \leftarrow L_k^* \forall k \in K$ (using pre-processing P1 if necessary).

The formulation of DBR with protection (e.g., dedicated or shared) follows similar approaches presented in the preceding discussions. For details of these formulations and the computations of link capacities for the case of accurate static demand, see R. D. Davis, K. Kumaran, G. Liu, and I. Saniee, "SPIDER: A Simple and Flexible Tool for Design and Provisioning of Protected Lightpaths in Optical Networks," Bell Labs Technical Journal, vol.6, no.1 (Jun. 2001), the disclosure of which is hereby incorporated by reference.

20

Routing With Resource Management

This section presents an enhancement to the basic shortest path routing and design-based routing through the application of trunk reservation for managing bandwidth resources during congestion. It is well known that trunk reservation may be applied to two-hop paths to reduce resource usage. Here, the trunk reservation concept is generalized to an arbitrary network with CSPF or DBR routing.

25

An enhancement to the basic DBR scheme, called adaptive DBR, is intended to deal with inaccuracies in traffic demand forecast. A main idea of adaptive

DBR is to use the current measured load for a given ingress-egress node pair as a surrogate to DBR. When the current measured load for a given node pair is lower than that used in the design of the DBR path, a connection request for the node pair will attempt to use the path given by DBR. On the other hand, when the current measured load is higher than that used in the design, a connection request can only use the path computed by DBR provided that the DBR path passes the trunk reservation test. In each case, a failed DBR path can fall back to CSPF with trunk reservation.

It is noted that adaptive DBR in an illustrative embodiment requires an online measurement component at each source node and an online path-selection component. For simplicity, the dynamic model with no protection is considered. The relevant notations will first be defined. Let v_{sd} be the designed load between s and d that is used in the computation of DBR paths. Let \hat{v}_{sd} be the current measured load between s and d .

Path selection for an exemplary adaptive DBR is shown in FIG. 6. When the designed load is greater than the current measured load, then the network is pruned using a trunk reservation of zero. If a DBR path exists, the DBR connection is setup (if possible). If no DBR path exists, then the network is pruned with a trunk reservation of r . A network is pruned with trunk reservation level r (i.e., $TR=r$) if any link with available bandwidth less than or equal to r is removed. The shortest path is computed on the pruned network, and the shortest path is setup if possible. When the designed load is less than or equal to the current measured load, then the network is pruned using a trunk reservation of r . If a DBR path exists, the DBR connection is setup (if possible). If no DBR path exists, then the network is again pruned with a trunk reservation of r . The shortest path is computed on the pruned network, and the shortest path is setup if possible.

Adaptive DBR is intended to improve upon offline DBR when the traffic demand forecast is inaccurate. If the demand is accurate, then adaptive DBR is generally the same as offline DBR.

Now, a technique called CSPF with trunk reservation (CSPF_TR) is described. The CSPF_TR is shown in FIG. 7. Let $l_{min}(s,d)$ denote the minimum

path length from source s to destination d computed through SPF. Then, the CSPF_TR scheme proceeds in a manner indicated in FIG. 7. The network is pruned with a trunk reservation of zero. The shortest path on the pruned network is determined. If the resulting shortest path is equivalent to $l_{min}(s,d)$, then the path is set up (if possible). If the path length is not equivalent to $l_{min}(s,d)$, then the network is pruned by using a trunk reservation of r . The shortest path on the pruned network is determined and the path is setup if possible.

The CSPF_TR of FIG. 7 provides short connections generally when network utilization is low, but also uses less resources than CSPF when network utilization is high. In particular, CSPF tends to use large amounts of resources when network utilization is high and CSPF_TR provides a better use of resources at high network utilization.

It is to be understood that the embodiments and variations shown and described herein are merely illustrative of the principles of this invention and that various modifications may be implemented by those skilled in the art without departing from the scope and spirit of the invention. The various assumptions made herein are for the purposes of simplicity and clarity of illustration, and should not be construed as requirements of the present invention.